# A Novel Path Error Compensation Method for Robotic Milling by Using Hybrid Temporal Network

Qingyu Peng, Wenlong Li, *Member, IEEE*, Cheng Jiang, Yuqi Cheng, Dongfang Wang, and Wei Xu, *Member, IEEE*

***Abstract*—In robotic milling, the path accuracy of the industrial robot is a crucial factor in ensuring the contour accuracy. Most existing compensation methods primarily focus on the single-point accuracy of industrial robots while neglecting the continuity of the milling path. This poses challenges to ensure the contour accuracy of the workpiece. Therefore, this article proposes a novel path error compensation method for robotic milling, first introducing a hybrid temporal network (HTN) to predict path error and then using a path iterative compensation (PIC) method to compensate for these errors. The HTN processes spatial information of milling path through convolution operations with error similarity of the industrial robot. Meanwhile, the temporal relationship of milling path can be extracted by temporal layers. Moreover, a dataset augmentation method is proposed to ensure the HTNs performance by comparing the sequence dispersion of each training milling path. Based on the HTNs prediction results, the PIC method is proposed to improve the final path accuracy by iterative compensation. To show the feasibility of the proposed method, a series of experiments are conducted. It is proved that the contour error of skin parts is within $\pm$0.35 mm.**

***Index Terms*—Error compensation, error prediction, neural network, robotic milling.**

## I. Introduction

SKIN parts are key components in aerospace field, where manufacturing accuracy significantly influences the aerodynamics and internal safety of aircraft. Due to the large scale of skin parts, most skin parts are difficult to machine with traditional computer numerical control machine tools [1]. Due to the flexibility of industrial robots, numerous companies and research institutes have explored robotic machining in the aviation industry [2], [3], [4]. In the field of skin part milling, as shown in Fig. 1, a typical robotic milling system mainly contains an industrial robot, a spindle (mounted on the end flange of the robot), and a control system [5], [6]. The robotic milling system generates the milling path based on target parts and the control system drives the robot to execute the preset milling path. However, due to the low positioning accuracy of the robot, the path accuracy of the robotic milling system cannot meet the requirement for the average contour error within $\pm$0.5 mm in edge milling of aircraft skin parts [7]. Therefore, improving the path accuracy of industrial robot is a crucial preparatory work for robotic milling.




Fig. 1. Automatic robotic milling [4].

Over the past few decades, many literature reports have focused on improving the absolute positioning accuracy of industrial robots. These studies are typically divided into model-based and model-free compensation methods. These model-based compensation methods involve obtaining positional error at several locations using a piece of measurement equipment and identifying the robot model parameter error based on the established model [8]. Common models include the standard Denavit–Hartenberg (D-H) model [9] and some modified models, including the modified D-H model [10], and the product-of-exponential (POE)-based model [11]. These methods can accurately describe the kinematic model of the robot, but they fail to account for the error caused by nonkinematic factor. In addition, some researchers considered the nonkinematic factors of the robot. The authors in [12] and [13] considered the stiffness

Received 27 February 2024; revised 26 June 2024 and 18 August 2024; accepted 8 October 2024. Recommended by Technical Editor A. Wahrburg and Senior Editor C. Clevy. This work was supported in part by the National Natural Science Foundation of China under Grant 52188102, Grant 52090054, Grant 52075203 and Grant 52205524, and in part by Hubei Province Key Research and Development Plan under Grant 2020BAA025. *(Corresponding author: Wenlong Li.)*

The authors are with the State Key Laboratory of Intelligent Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: 83524374@qq.com; wlli@mail.hust.edu.cn; 307080666@qq.com; chengyuqi.c@qq.com; 978427099@qq.com; weixu.chn@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMECH.2024.3479410.
Digital Object Identifier 10.1109/TMECH.2024.3479410

models of the industrial robot. The proposed stiffness model can compensate for minor deformations caused by the robot's low stiffness, thereby improving machining accuracy. To ensure the high accuracy for both side milling and climb milling, the authors in [14], [15], and [16] addressed the reversal error in the robot's kinematic model. The reversal error indicates that when running the same path in forward and reverse directions, the path error of the robot is not identical. This also suggests that the reversal error can be reduced. To solve for the calibration parameters of the model, common methods, such as the least square method [17], Levenberg–Marquardt (L-M) method [18], and the extended Kalman filter (EKF) method [19], can be used. These methods can achieve satisfactory results in solving calibration parameters. However, robots with external loads are commonly used in robotic milling systems to meet the requirements of milling force and complex tasks. Unlike light-load robots, these robots have joint deformation and friction due to the end-effector's load and motors' influence. These factors exhibit a high nonlinearity, making them challenging to model accurately and solve quickly [20]. Therefore, many scholars have researched model-free compensation methods for industrial robots to handle the problem of nonlinear error.

These model-free compensation methods avoid the problem of nonlinear error by learning a complex nonlinear mapping relationship between error and joint angles through a data-driven approach. The interpolation method is a general model-free compensation method. The authors in [21] and [22] proposed an interpolation method based on known data points. To increase solving efficiency and reduce redundant information, Zeng et al. [23] established an optimal estimation model with fewer known data points by analyzing the similarity between joint angles. This model demonstrates that effective known data points can help improve the positioning accuracy. Meanwhile, some scholars made improvements to the interpolation method itself. Chen et al. [24] used the cross-half difference function to explain the relationship between joint angles and positioning error. Min et al. [25] proposed a kriging-based error compensation method to improve the positioning accuracy of the robot.

These interpolation methods improve the positioning accuracy of the industrial robot to a certain extent. However, due to the high nonlinearity of industrial robots, these interpolation methods cannot accurately predict the absolute positioning error of the robot with limited fitting capability. Motivated by the excellent generalization and nonlinear mapping capability of neural networks, some scholars used neural networks to improve the positioning accuracy of robots. The authors in [26] and [27] used a neural-network-based method to predict the nongeometric error of robots. However, the performance of neural networks is highly dependent on the selection of hyperparameters. Choosing and tuning these hyperparameters is also a significant challenge. Some scholars have conducted research on the initialization of neural-network hyperparameters by different intelligent optimization algorithms. Li et al. [28] used a genetic particle swarm algorithm to optimize the neural network. Cao et al. [29] used a neural network based on a butterfly and flower pollination algorithm to improve the robot's absolute pose accuracy. Meanwhile, due to the ability of deep networks to extract high-level features from data, some scholars have used deep neural networks to improve the positioning accuracy of robots. Yang et al. [30] proposed an EKF-incorporated residual neural network to improve the calibration accuracy. Zhao et al. [31] constructed an automatic measuring system to collect more than 10 000 sampling points for training the deep neural network. In addition to deepening the network structure, some scholars have also considered integrating the characteristics of robotic machining with the neural network. Wang et al. [32] proposed a deep belief network, which considered the error similarity of the robot, showing better prediction accuracy and inference speed. Sun et al. [33] used the long short-term memory (LSTM) network and residual convolutional network to forecast the tool condition for improving the milling quality indirectly. Considering the temporal feature of the milling path, Bilal et al. [34] improved the accuracy of robotic milling by using the LSTM network to predict the path error of the industrial robot.

Existing neural-network methods mostly improved the single-point positioning accuracy of industrial robots, enabling its application in aviation manufacturing. However, the correlation among joint configurations in milling paths and the temporal features of the milling path has not been fully considered. The objective of this article is to enhance the robot's path accuracy by leveraging both the temporal features of the milling path and the spatial correlations among joint configurations. Therefore, this article proposes a novel path error compensation method, first proposing hybrid temporal network (HTN) to predict path error and then using path iterative compensation (PIC) method to compensate for predicted path error. The HTN combines multilayer convolutional neural networks (MCNNs) layers and an LSTM layer. Based on the error similarity, the MCNN layer extracts spatial information of joint configuration to acquire high-level joint semantic information. Following an MCNN layer, an LSTM layer extracts the long-term dependencies between the sequences to obtain the tendency of the path error. Furthermore, due to the scarcity of real dataset, a dataset augmentation method is proposed to ensure the network's performance. This method selects appropriate sequences based on Kullback–Leibler (KL) sequence dispersion. In addition, PIC method is used to compensate path error. Based on the predicted path error, the PIC method is used to input modified path points into the controller by iterative compensation. In summary, the main contributions are listed as follows.

1) Propose a novel compensation method to improve the robot's path accuracy. First, HTN extracts spatial information of each joint configuration and captures the variation trend of path error. Subsequently, based on the high prediction accuracy of the HTN, PIC method improves the robot's path accuracy.
2) Propose a data augmentation method to improve network performance. The proposed dataset augmentation method selects appropriate training sequences based on KL sequence dispersion. This method effectively ensures the performance of the network.
3) Extensive experiments demonstrate the effectiveness of the proposed method. Path compensation experiments

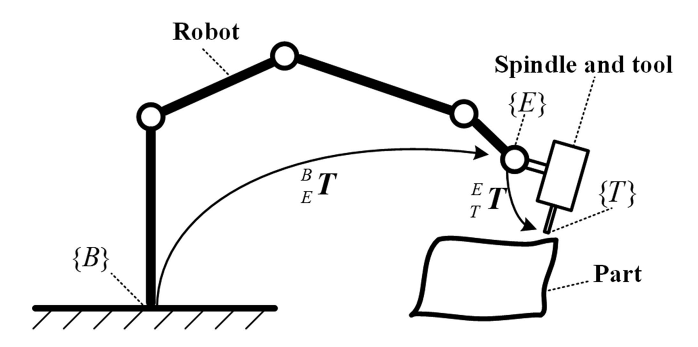



Fig. 2. Geometry structure of robotic milling system.

verified the effectiveness of the proposed method. Milling experiments confirm that the proposed compensation method can be applied to practical skin part milling scenarios, with contour accuracy meeting the requirement of ±0.5 mm.

The rest of this article is organized as follows. Section II analyzes the robot's error and HTN. Section III discusses the dataset augmentation method and the PIC method. The practical experiments validate the effectiveness of the proposed compensation method in Section IV. Finally, Section V concludes this article.

## II. Robot Path Error Prediction Network

The single-point positioning error is determined by the robot's joint angle configuration, while the robot's path points possess corresponding spatial and temporal characteristics. The HTN enhances prediction accuracy by extracting spatial information and sequential temporal relationships.

### A. Error Analysis of Industrial Robots

As shown in Fig. 2, $\{B\}$ and $\{E\}$ represent the basic coordinate system and the end-flange coordinate system of the industrial robot. $\{T\}$ rigidly connected to $\{E\}$, represents the coordinate system of the spindle and tool. The homogeneous transformation matrix from $\{T\}$ to $\{B\}$ can be described as follows:

$$ {}_{T}^{B}\boldsymbol{T} = {}_{E}^{B}\boldsymbol{T}{}_{T}^{E}\boldsymbol{T} \tag{1}$$

where ${}_{n}^{m}\boldsymbol{T}$ denotes a homogeneous transformation matrix from $\{n\}$ to $\{m\}$.

The transformation matrix ${}_{E}^{B}\boldsymbol{T}$ represents the kinematic model of the six-axis industrial robot. From the kinematic model, if only the geometric error is considered, the pose error matrix can be described by the differential of ${}_{E}^{B}\boldsymbol{T}$

$$\begin{aligned} \Delta_{E}^{B}\boldsymbol{T} &= \boldsymbol{f}(\boldsymbol{a},\boldsymbol{\alpha},\boldsymbol{d},\boldsymbol{\theta}) \\ &= \prod_{i=0}^{5}\left({}_{i}^{i+1}\boldsymbol{T} + \frac{\partial_{i}^{i+1}\boldsymbol{T}}{\partial a_i}\Delta a_i + \frac{\partial_{i}^{i+1}\boldsymbol{T}}{\partial \alpha_i}\Delta \alpha_i + \frac{\partial_{i}^{i+1}\boldsymbol{T}}{\partial d_i}\Delta d_i \right. \\ &\quad \left. + \frac{\partial_{i}^{i+1}\boldsymbol{T}}{\partial \theta_i}\Delta\theta_i\right) \end{aligned} \tag{2}$$

where $\boldsymbol{a},\boldsymbol{\alpha},\boldsymbol{d},$ and $\boldsymbol{\theta}$ are the vector representation of the D-H parameters, $\Delta a_i, \Delta\alpha_i, \Delta d_i,$ and $\Delta\theta_i$ represent the minor error in the D-H parameters of each link, and $\Delta_{E}^{B}\boldsymbol{T}$ is the pose error matrix (geometric error) of the industrial robot.

However, in robotic milling, the self-weight and external load of the industrial robot can cause flexible deformation of each joint angle [35]. The nonlinear deformation in these joint angles results in additional nonlinear error in the robot. Therefore, in a six-axis industrial robot, the positioning error can be described as follows:

$$ f(\boldsymbol{a},\boldsymbol{\alpha},\boldsymbol{d},\boldsymbol{\theta}')_i = f_n + f_e \tag{3}$$

where $\boldsymbol{\theta}'$ represents the six-axis joint angle vector of the robot in actual machining state (joint angles are affected by self-weight and external load). $f_n \in R^{3\times1}$ represents the geometric error and it can be obtained by extracting the last column of the $\Delta_{E}^{B}\boldsymbol{T}$. $f_e = [f_{ex} \quad f_{ey} \quad f_{ez}]^{\boldsymbol{T}} \in R^{3\times1}$ represents the nongeometric error. $f_{ex}, f_{ey},$ and $f_{ez}$ are the positional errors corresponding to the joint vector (caused by nongeometric) along the $x, y,$ and $z$ axes, respectively. Among them, $f_{ex}$ (the same applies to the other two items) can be expressed as follows:

$$\begin{aligned} f_{ex} &= \sum_{i=1}^{6} f_i(\theta_1,\theta_2,\ldots,\theta_6) \\ & f_i(\theta_1,\theta_2,\ldots,\theta_6) \\ &= \sum_{i=1}^{6} A_n(s\boldsymbol{\theta}_1)^{n_{1i}}(c\boldsymbol{\theta}_1)^{n_{2i}}\ldots(s\boldsymbol{\theta}_6)^{n_{6i}}(c\boldsymbol{\theta}_6)^{n_{6i}} \end{aligned} \tag{4}$$

where $A_n$ and $n_{1i}, n_{2i},\ldots,n_{6i}$ are all constant terms, $\theta_1,\theta_2,\ldots,\theta_6$ represent the six-axis joint angles, and $s(\bullet)$ and $c(\bullet)$ are all basic elementary functions, which are continuous in the set of real numbers. From (4), it is observed that the configuration of robot's joint angle (the combination of joint angles for six joints) determines the positioning error of the industrial robot at different locations so that the spatial interrelationships among the six joint angles should be taken into account. Furthermore, the milling path consists of a series of continuous path points. The correlation between these path points should be considered. An error similarity exists [24], [32] between neighboring path points within a specific geometric range, resulting in positional error similarity among these path points. In addition, the robot executes a preset milling path, thereby establishing dependencies between each path point [34] (we refer to these dependencies as temporal features in this article). These correlations reflect that the robot's positioning error exhibits a distinct trend in response to the robot's joint configuration change. Therefore, as shown in Fig. 3, utilizing the spatial information of joint configuration and the temporal relationship of the consecutive path points are crucial factors in improving the prediction accuracy of the HTN.

### B. HTN to Predict Path Error

Based on those mentioned above, as shown in Fig. 4, HTN processes spatial information among each path point through convolution operations. Meanwhile, the temporal relationship

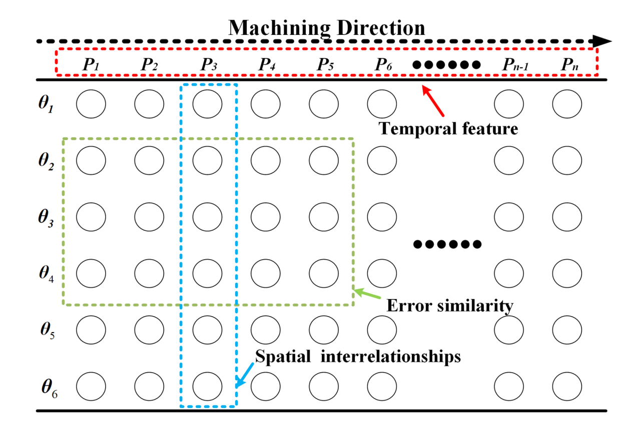



Fig. 3. Feature factors of the milling sequence.

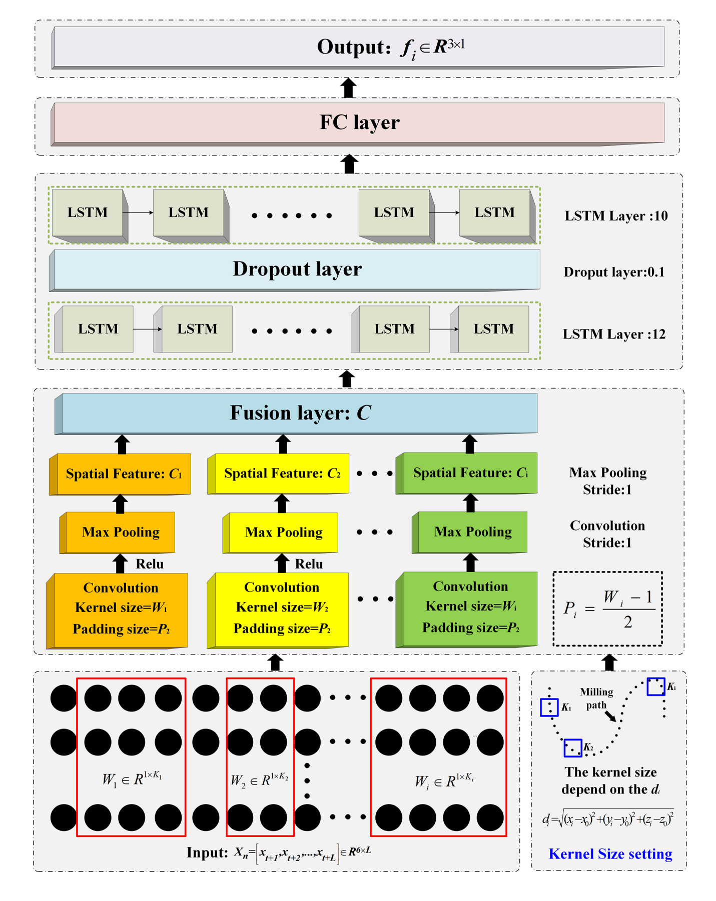



Fig. 4. Main structure of HTN.

can be extracted through recursive processes. The input of HTN is a multichannel sequence (a series of robot's joint configurations), which can be described as follows:

$$\boldsymbol{X}_n = [x_{t+1}, x_{t+2}, \ldots, x_{t+L}] \quad (5)$$

where $\boldsymbol{X}_n \in R^{6\times L}$ represents the $n$th input sequence, and $L$ means the batch size of input. The MCNN layer is adopted in the first phase to capture the spatial information among each path point. The LSTM layer is used to extract the sequential temporal relationship. Similarly, the output corresponds to the robot's positioning error for each path point $f_i \in R^3 (i = 1, 2, \ldots, L)$.

*1) MCNN Layers:* Extracting spatial information from the multichannel sequence is crucial for prediction. In the input of network, spatial features encompass the distribution of joint angles between each channel and the variations between neighboring path points (joint angle configurations). Due to the convolutional neural networks' ability to indirectly capture interactions and dependencies between independent elements [37], the MCNN layer is designed to extract spatial features to acquire higher level spatial feature information. Specifically, convolution operations are applied to neighboring path points at each time step of the multichannel sequence. This operation effectively captures the high-level joint semantic information of the robot. This high-level joint semantic information includes the relationships among the six joint angles of the robot, as well as the semantic information related to the variations in joint angles between neighboring path points. They jointly illustrate the collaborative relationship among joint errors in the robot. Therefore, the convolution operation can be described as follows:

$$\boldsymbol{C}^{(k)} = \boldsymbol{W}^{(k)} * \boldsymbol{X}_{t-L:t-1} \quad (6)$$

where $\boldsymbol{W}^{(k)}$ is the convolution kernel, and $\boldsymbol{C}^{(k)}$ is the feature extracted from the relationship between variables. Meanwhile, based on the principle of error similarity, robot positioning error exhibits similarity within the same region (i.e., in Cartesian space, within a certain geometric range, robot positioning error exhibits similarity). Hence, multiple convolution kernels are utilized for repeated extraction to extract more local similarity information.

As shown in Fig. 4, the kernel sizes are also set differently based on the error similarity of the robot. Set $P_0 = (x_0, y_0, z_0)$ is the theoretical coordinate of the robot's target position. The distance between the other path points and the theoretical coordinate of the robot's target position can be described as follows:

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} < \varepsilon \quad (7)$$

where $\varepsilon$ is the maximum allowed distance. If $d_i < \varepsilon$, the position error of the $K$ sample in the robot workspace is similar to the target point. In order to extract the error similarity information of path points, multiple convolutions are employed to extract data from these path points. The size of the convolution kernel corresponds to the number of neighboring path points with error similarity. Due to the complexity of the milling path, the count of neighboring path points satisfying the error similarity principle varies within a fixed error similarity range for different path points. In addition, the number of neighboring points satisfying the error similarity principle is computed for each path point and denoted as $K_i (i = 1, 2, 3, \ldots, 6)$. With a size of $K_i (i = 1, 2, 3, \ldots, 6)$, dimensional convolution kernels are used to perform convolution on the milling sequence individually, thereby comprehensively extracting the error similarity features of the sequence. Finally, to adjust the kernel size and padding size, the dimensions of the extracted spatial features remain consistent.

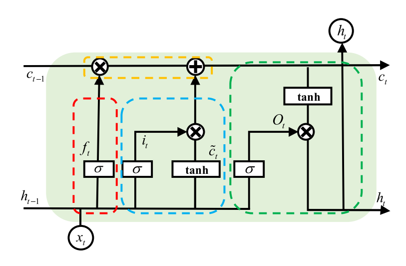



Fig. 5. Diagram of LSTM unit.

By integrating these diverse spatial features, the spatial feature input to the subsequent LSTM layer is obtained. The fused function can be written as follows:

$$C = \omega_1 \cdot C^{(1)} + \omega_2 \cdot C^{(2)} + \cdots + \omega_k \cdot C^{(k)} \tag{8}$$

where $\omega_1, \omega_2, \ldots, \omega_k$ are the learnable weight coefficients and $C$ is the final fusion feature.

*2) LSTM Layers:* Capturing the sequential temporal relationship of the machining path is the key factor to obtain the tendency of the path error. As the multichannel sequence is typically a long machining sequence, traditional temporal networks may encounter issues, such as gradient vanishing and exploding, when dealing with long sequence tasks. The LSTM can avoid gradient disappearance and explosion with cell units. As shown in Fig. 5, the cell unit of LSTM consists of three gates, including a forget gate, an input gate, and an output gate. The forget gate is adopted to decide what information will be discarded. The input gate is used to decide how much information is added to the cell state at the current time. The output gate is adopted to control the information of the cell state at the current time as the output [37]. The detailed formulae of LSTM cell units from the input to the output are obtained as follows:

$$\begin{aligned}
f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)\\
i_t &= \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)\\
\overline{c_t} &= \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right)\\
c_t &= f_t \cdot c_{t-1} + i_t \cdot \widetilde{c_t}\\
o_t &= \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right)\\
h_t &= o_t \cdot \tanh\left(c_t\right)
\end{aligned} \tag{9}$$

where $f_t, i_t,$ and $o_t$ represent the forge gate, input gate, and output gate, respectively. $\overline{c_t}$ represents the candidate vector and $c_t$ represents the state of the cell. $W_f, W_i, W_c,$ and $W_o$ represent the coefficient matrices that need to be learned. $b_f, b_i, b_c,$ and $b_o$ represent the bias vectors. $\sigma(\cdot)$ and $\tanh(\cdot)$ represent the activation functions.

Due to the network's ability to capture dependencies over longer time intervals in sequential data, the LSTM layer decodes the fusion future from the MCNN layer by extracting the long-term temporal relationship. Finally, the output from the LSTM layer is fed into the fully connected (FC) layers, and the FC layer is employed to produce the robot's ultimate predicted path error result through regression.

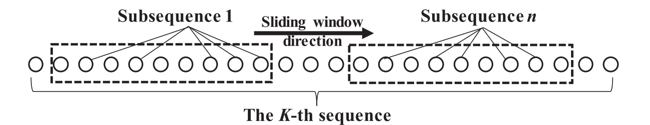



Fig. 6. Milling points classification.

---

**Algorithm 1:** Dataset Augmentation Method.

---

**Input**: Initial set of $K$ sequence: $\{X_1^{cl}, X_2^{cl}, X_3^{cl}, \ldots, X_k^{cl}\}$.

**Output**: Augmented dataset: $\{X_1, X_2, \ldots, X_n\}$.

1. Set the window length $L$ and window movement interval $I$. Then, make windows slide sequentially in the initial set of $K$ sequence.
2. Set two empty lists: $C$ and $D$.
3. In each window, the linear congruence method is used to randomly select discrete path points and arrange them into sequences in the same order:
4. For $i = 0$ to max iterations
5. Evaluate each subsequence's sequence dispersion (SED) and add the subsequence to the list $C$.
6. End
7. Select the specific subsequence by the SED from list $C$ and add it to the list $D \in \{X_1, X_2, \ldots, X_n\}$.

---

## III. Preparations for Error Compensation

A dataset augmentation method is initially employed to obtain a trained HTN with high performance. The Beluga whale optimization (BWO) method is used for the optimal initial parameters for network training. Based on the trained HTN, the PIC method can enhance the robot's path accuracy with high prediction accuracy of HTN.

### *A. Datasets Augmentation for Network Training*

Due to the scarcity of the robot milling dataset, a dataset augmentation is proposed to ensure the network's well performance under various milling conditions. The most commonly used methods for dataset augmentation are flipping, scaling, and windows warping. In robotic milling, flipping and scaling may disrupt the actual path error of the industrial robot, leading to a negative impact. Windows warping can be used for network training by adding diverse milling sequences.

Note that the executable path points dispersed by the robot's control system are approximately equidistant. It is observed that the dispersion of each sequence is almost uniform (the distance between each path point is equal). Based on the windows warping and sequence dispersion, a dataset augmentation method for network training is proposed to make the milling points as equidistant as possible. All path points are initially divided into $K$ sequences by *K-means* clustering method. For each sequence, as shown in Fig. 6, subsequences are randomly generated from each sequence by appropriate window length and pseudorandom numbers. These subsequences are filtered to

the datasets by evaluating the degree of dispersion. This method can be described as follows:

As the standard equidistant milling sequence consists of equidistant path points, it exhibits the lowest degree of dispersion. The sequence dispersion can be obtained by comparing it to the standard sequence. Therefore, some concepts of information theory are used to evaluate the sequence dispersion of each sequence.

In information theory, the KL sequence dispersion can measure the distance between two random distributions. When the difference between two random distributions increases, the relative entropy increases

$$\begin{aligned}\mathrm{KL}(q||p) &= \sum_{i=1}^{N} q(x_i) \cdot (\log q(x_i) - \log p(x_i)) \\ &= \sum_{i=1}^{N} q(x_i) \cdot \log \frac{q(x_i)}{p(x_i)}\end{aligned} \tag{10}$$

where $p(x_i)$ and $q(x_i)$ represent the different probabilities of two different probability distributions. When the probability distribution of the selected sequence is known, the KL sequence dispersion can be used to compare the sequence dispersion. The expression form of "probability" should be defined to obtain the probability distribution of the selected sequence. The Euclidean distance can be expressed as follows:

$$d_i = f(\boldsymbol{\theta}_i) - f(\boldsymbol{\theta}_{\mathbf{i}+1}) \tag{11}$$

where $\boldsymbol{\theta}_i$ represents a theoretical joint angle and $d_i$ describes the corresponding Euclidean distance. The probability of each path point can be defined as follows:

$$\begin{aligned}P\{i=k\} = p(k) &= \frac{\exp(f(\boldsymbol{\theta}_k) - f(\boldsymbol{\theta}_{k+1}))}{\sum \exp(f(\boldsymbol{\theta}_k) - f(\boldsymbol{\theta}_{k+1}))} \\ P &= \sum_{k=1}^{j-1} p(k) = 1\end{aligned} \tag{12}$$

where $p(k)$ represents the probability of each path point, and $P$ represents the probability distribution of each milling path. Based on the expression form of a probability distribution, the probability distribution of the standard sequence can be expressed as follows:

$$\begin{aligned}Q\{i=k\} = q(k) &= 1/L_k \\ Q &= \sum_{k=1}^{j-1} q(k) = 1\end{aligned} \tag{13}$$

where $q(k)$ represents the probability of each path point, and $Q$ represents the probability distribution of equidistant distribution. The difference between the unknown distribution $P$ and the known distribution $Q$ can be expressed as follows:

$$\begin{aligned}\mathrm{KL}(Q\,\|P) &= \sum_i Q(x_i) \log \frac{1}{P(x_i)} - \sum_i Q(x_i) \log \frac{1}{Q(x_i)} \\ &= \sum_{i=1}^{L_k-1} q(x_i) \ln L_k - \sum_{i=1}^{L_k-1} q(x_i) \ln \frac{1}{q(x_i)} \\ &= \ln \sum_{i=1}^{L_k-1} e^{(f(\boldsymbol{\theta}_i) - f(\boldsymbol{\theta}_{i+1}))} \\ &\quad - \frac{1}{L_k} \sum_{i=1}^{L_k-1} (f(\boldsymbol{\theta}_i) - f(\boldsymbol{\theta}_{i+1})) - \ln L_k\end{aligned} \tag{14}$$

where the third term of (14) is constant. As the constant term does not affect the equation, the constant term can be removed. Therefore, the middle part can be recorded as an effective factor, which can be defined as SED

$$\mathrm{SED} = \ln \sum_{i=1}^{L_k-1} e^{(f(\boldsymbol{\theta}_i) - f(\boldsymbol{\theta}_{i+1}))} - \frac{1}{L_k} \sum_{i=1}^{L_k-1} (f(\boldsymbol{\theta}_i) - f(\boldsymbol{\theta}_{i+1})) \tag{15}$$

where the first term is the variable of $e$, and the second is the arithmetic mean of $f(\boldsymbol{\theta}_i) - f(\boldsymbol{\theta}_{i+1})$. Note that the numerical size of (15) represents the probability of the sequence being selected. If the sequence gains a smaller SED, it will have a higher chance of being selected. Thus, by estimating the numerical size of SED, the dataset can be created to enrich the data diversity. With the diverse dataset, the performance of HTN can be guaranteed.

### B. Training Procedure

Once the appropriate dataset is available, HTN can be trained. To evaluate the prediction accuracy of the network, the mean square error between the expected output of models $h(t)$ and the actual output $y(t)$ is taken as the fitness function

$$X_{\text{best}} = \sum_{i=1} \sum_{j=1} \|h(t) - y(t)\|^2. \tag{16}$$

In the training process, the initialization of parameters is critical to the effectiveness and optimal performance of network training. To get better initial parameters for the network, the BWO method is used in this article to obtain the optimal initial weights of the network [38]. When the optimal initial parameters are reached, the adaptive moment estimation (Adam) optimizer with fast convergence [39] is used to optimize the loss function.

### C. PIC Method

Based on HTN with high performance, PIC method is presented in this section. The traditional direct compensation method achieves modified path points by directly adding the predicted error to the nominal error and feeding them into the robot controller. Due to the calibration error and inherent motion error of the robot, the controller cannot accurately guide the robot to the desired path points during the correction process. Before robotic milling, the PIC method is used to input path points (obtained from Algorithm 2) into the controller. The principle of the method is shown in Fig. 7, and the specific method is detailed in Algorithm 2.

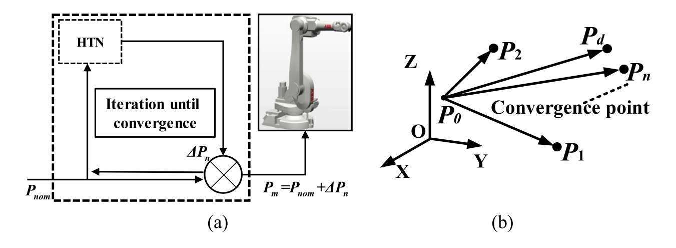



Fig. 7. PIC method. (a) Path compensation. (b) Iterative compensation.

**Algorithm 2:** A PIC Method.

**Input:** An initial path point $P_{\text{nom}}$

**Output:** A modified machining path point $P_i$

1. For $i = 0$ to max iterations $N$
2. Solve the joint angle $q_{\text{nom}}$ in position $P_{\text{nom}}$
3. Input the current desired joint angle into the network to get the predicted error $\Delta P_i$ of the path point.
4. Add the predicted error $\Delta P_i$ on the nominal position to obtain the modified path points $P_i = P_{\text{nom}} + \Delta P_i$.
5. Set the $P_{\text{nom}} = P_i$
6. End
7. Return modified machining path point $P_i$

Therefore, the predicted error $\Delta P_i$ can be represented as follows:

$$\Delta P_i = \text{HTN}\left(P_{\text{nom}} + \Delta P_{i-1} + \Delta P_{i-2} + \cdots + \Delta P_0\right)$$
$$\text{s.t} \quad \Delta P_0 = 0 \tag{17}$$

where $\text{HTN}(\cdot)$ represents the output of HTN and the modified path point $P_i = P_{\text{nom}} + \Delta P_i$, which can be inputted into the controller to compensate the path error. The proposed PIC method in this article involves using the trained HTN to predict and correct path error repeatedly. Due to the high prediction accuracy of the trained HTN (analogous to a laser tracker), this process ultimately brings the robot closer to the ideal path points. Specifically, when the robot receives a motion command, it must move from the current position $P_0$ to the target position $P_d$. The initially prescribed displacement is set as the predicted value $x_0$ from the HTN. However, due to the prediction error, the actual displacement becomes $x_0 + e_0$. Meanwhile, taking into account the calibration error and nongeometric factors, we denote them as $\boldsymbol{R}_{\text{err}}$ (a rotation matrix). The robot's actual motion can be represented as follows:

$$x_1 = (\boldsymbol{I}_3 - \boldsymbol{R}_{\text{err}})(x_0 + e_0). \tag{18}$$

Similarly, when progressing to the $N$th iteration, the actual motion of the robot can be expressed as follows:

$$x_n = (\boldsymbol{I}_3 - \boldsymbol{R}_{\text{err}})^n x_0 + (\boldsymbol{I}_3 - \boldsymbol{R}_{\text{err}})^n e_0 + \cdots$$
$$+ (\boldsymbol{I}_3 - \boldsymbol{R}_{\text{err}}) e_n \tag{19}$$

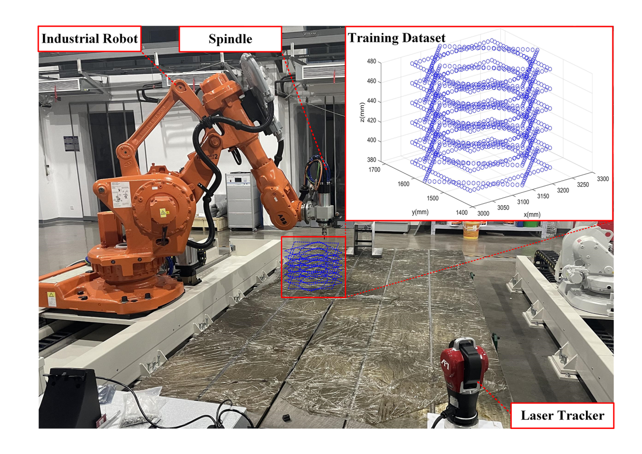



Fig. 8. Experimental platform and training dataset.

where $R_{\text{err}} \in \text{SO}(3)$ represents the calibration error and nongeometric factors. Taking the norm of $(\boldsymbol{I}_3 - \boldsymbol{R}_{\text{err}})^n$ [40], it can be obtained as follows:

$$\|(\boldsymbol{I}_3 - \boldsymbol{R}_{\text{err}})^n\|_2^n \le \boldsymbol{U}^H$$
$$\times \left\| \begin{pmatrix} (1-\cos\alpha + i\sin\alpha)^n & 0 & 0 \\ 0 & (1-\cos\alpha - i\sin\alpha)^n & 0 \\ 0 & 0 & 1 \end{pmatrix} \right\|_F^2$$
$$\times \|\boldsymbol{U}\|_2^2$$
$$= \left(|(1-\cos\alpha + i\sin\alpha)|^{2n} + |(1-\cos\alpha - i\sin\alpha)|^{2n}\right)$$
$$= 2\left(\sqrt{(1-\cos\alpha)^2 + (\sin\alpha)^2}\right)^{2n}$$
$$= 2\left((1-\cos)^2 + (\sin\alpha)^2\right)^n$$
$$\le 2(2 - 2\cos\alpha)^n \tag{20}$$

where $\|\centerdot\|_F$ represents the Frobenius norm. To ensure the convergence of $(\boldsymbol{I}_3 - \boldsymbol{R}_{\text{err}})^n$, it is only necessary to set $|2 - 2\cos\alpha| < 1$, meaning that the absolute value of the rotation angle of $\boldsymbol{R}_{\text{err}}$ should be less than 60°. This demonstrates the convergence of the Algorithm 2. As indicated by (18), the specific value of $\boldsymbol{R}_{\text{err}}$ is not necessary to know. Therefore, when the prediction error $e_i$ is sufficiently small, the sequence $x_n$ guided by the trained HTN can achieve convergence.

## IV. Experiment and Practical Application

### A. Experimental Platform

As shown in Fig. 8, an experimental platform is developed to verify the effectiveness of the proposed method. The platform consists of an ABB IRB 6660-1.93/205 industrial robot with a spindle, an API Radian laser tracker with a target ball, and other equipment. The spindle mounted on the robot's flange is used for milling the aircraft skin sample part. The API Radian laser tracker is used as a sensor to track the actual error of the industrial robot. The parameters of the laser tracker are shown in Table I. The path compensation experiments

TABLE I
PARAMETERS OF API LASER TRACKER

| Parameter | Value |
|---|---|
| Range of measuring | 10m |
| Accuracy | $10\mu m + 5\mu m$ / m |
| Repeatability | 1 ppm |

TABLE II
PARAMETERS OF THE PROPOSED MODEL

| Description | Value |
|---|---|
| Activation function | ReLU |
| Optimization algorithm | Adam |
| Loss function | MSE |
| Initial learning rate | 0.01 |
| MSE of x-direction | 0.0272mm |
| MSE of y-direction | 0.0640mm |
| MSE of z-direction | 0.0426mm |
| Network training time | 600s |
| Network target threshold | 0.001mm |
| Number of network training epochs | 2000 |

and milling experiments are carried out with the experimental platform.

## B. Path Error Compensation Experiment

Before performing robot path compensation operations, HTN needs to be trained. Several common milling paths in the robot's workspace are planned. As shown in Fig. 8, these milling paths can be separated into 1280 sampling points, which can be used to train the network. In addition, to address the issue of inconsistent error between the robot's forward and reverse movements, a total of 2560 measurement data points (including both forward and reverse movements) are simultaneously collected (as side milling being commonly used in aviation skin milling and the robot's orientation changes are relatively small, our experiments have not considered situations in other directions). The input for HTNs training contains joint angle configuration sequences, which are obtained by the robot's inverse kinematics. The HTNs output is the error at each path point. To validate the effectiveness of the network training, we reserve 20% of the dataset as a validation set. Furthermore, mean-squared error (MSE) is utilized as a metric to analyze the effectiveness of the network training. MSE is a loss function, which calculates the mean of the squared differences between the predicted and actual values. The closer the MSE is to 0, the more consistent the model's predicted value is with the actual value. In addition, as shown in Table II, the specific parameters of the network, such as the MSE of the test set, training time, and initial learning rate, are all presented. It is observed that HTN has good adaptability and robustness for predicting the path error of industrial robots.

After completing the network training, we compensate for the robot's path. As shown in Fig. 9, based on the ISO 9283 standard, a test path is designed to verify the HTNs generalization capabilities. By driving the robot in both forward and reverse sequences along the test path, the effectiveness of different path error compensation methods is validated. The test path is

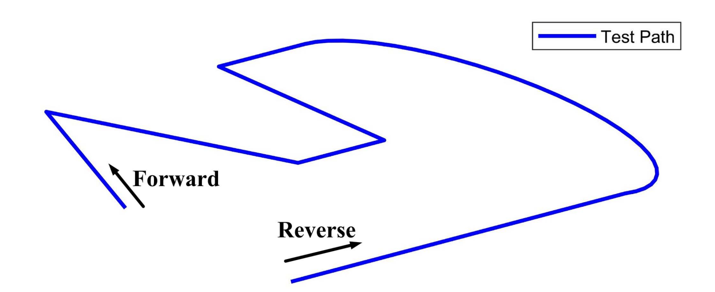



Fig. 9. Actual test path.

TABLE III
RELEVANT DETAILS OF COMPARISON METHODS

| Description | Method1 | Method2 | Method3 |
|---|---|---|---|
| Prediction model | DH-model | DNN | LSTM |
| Compensation method | Direct compensation | Direct compensation | Direct compensation |
| Optimization method | L-M method | Adam | Adam |
| Layer/Node | \ | [10], [12] | [15] |
| Training epochs | \ | 2000 | 2000 |

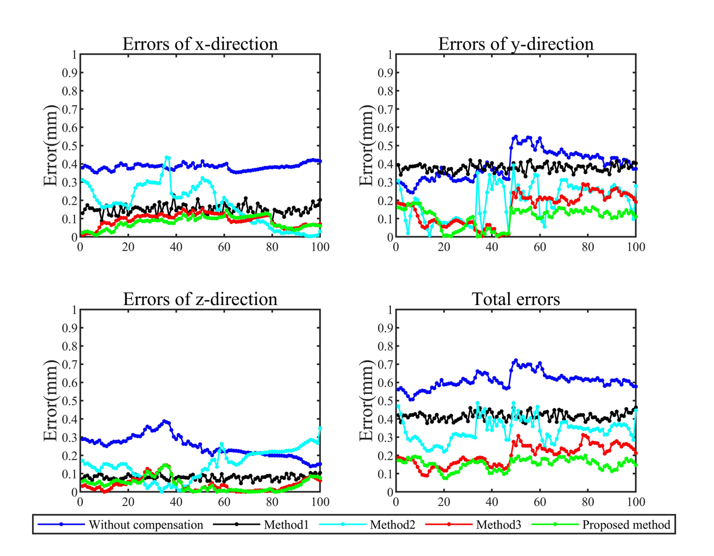



Fig. 10. Absolute value of path error of forward path.

approximately within an area of 400 mm $\times$ 400 mm on a flat plane, with the $z$-axis height 50 mm above the maximum height of the training dataset.

The laser tracker records the positions of path points at a certain frequency. By comparing these position errors with the theoretical path, the $x$-axis, $y$-axis, $z$-axis, and total positioning error of path points are obtained. These path error compensation experiments are conducted five times: four times using different path error compensation methods, including parameter calibration method [10] (method 1), DNN compensation method [31] (method 2), LSTM compensation method [34] (method 3), and the proposed method. The relevant details of comparison methods 1, 2, and 3 are shown in Table III. The results of the path error are shown in Figs. 10, 11 and Tables IV and V, referencing to the definition of

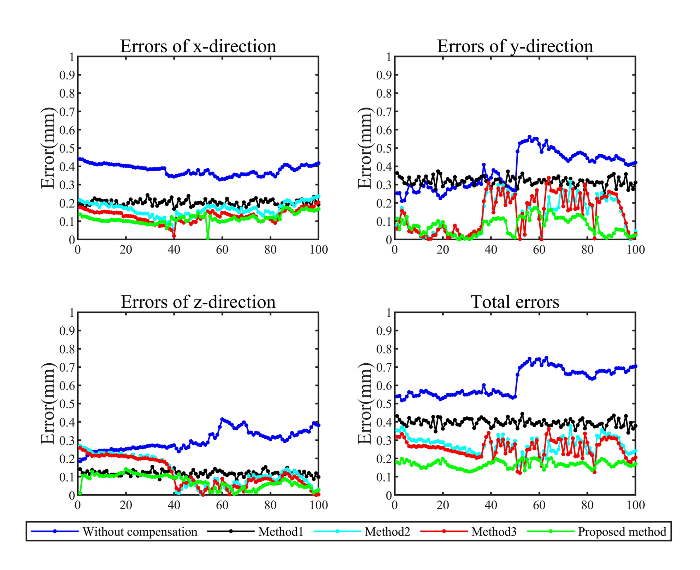



Fig. 11. Absolute value of path error of reverse path.

TABLE IV
AVERAGE ERROR OF FORWARD PATH (UNIT: MM)

| | Without compensation | Method1 | Method2 | Method3 | Proposed Method |
|---|---|---|---|---|---|
| *x*-axis | 0.3834 | 0.1502 | 0.1747 | 0.0920 | **0.0794** |
| *y*-axis | 0.3944 | 0.3785 | 0.1940 | 0.1544 | **0.1069** |
| *z*-axis | 0.2495 | 0.0793 | 0.1493 | 0.0406 | **0.0805** |
| Total | 0.6105 | 0.4157 | 0.3380 | 0.2014 | **0.1639** |

TABLE V
AVERAGE ERROR OF REVERSE PATH (UNIT: MM)

| | Without compensation | Method1 | Method2 | Method3 | Proposed Method |
|---|---|---|---|---|---|
| *x*-axis | 0.3800 | 0.2012 | 0.1561 | 0.1287 | **0.1157** |
| *y*-axis | 0.3801 | 0.3183 | 0.1387 | 0.1413 | **0.0950** |
| *z*-axis | 0.2973 | 0.1195 | 0.1193 | 0.1171 | **0.0735** |
| Total | 0.6207 | 0.3958 | 0.2785 | 0.2554 | **0.1879** |

TABLE VI
PATH ERROR OF FORWARD AND REVERSE PATH (UNIT: MM)

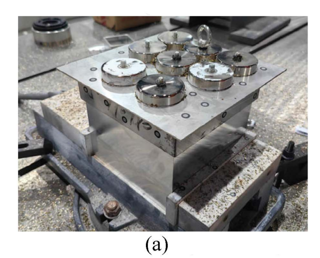
| | Without compensation | Method1 | Method2 | Method3 | Proposed Method |
|---|---|---|---|---|---|
| Forward | 0.7214 | 0.4632 | 0.3875 | 0.3124 | **0.1997** |
| Reverse | 0.7505 | 0.4441 | 0.3799 | 0.3619 | **0.2025** |

ISO-9283. The path accuracy is the maximum deviation between path points and the theoretical path points, i.e., $AT_P = \sqrt{(\overline{x_i} - x_{ci})^2 + (\overline{y_i} - y_{ci})^2 + (\overline{z_i} - z_{ci})^2}(i = 1, 2, \ldots, n)$.

By analyzing the results in Tables IV–VI and Figs. 10 and 11, it is observed that the total average error and path error of all routes using the compensation method are significantly lower than those not using the compensation method. The average compensated error is reduced within 0.5 mm, which shows that these methods have effectiveness and feasibility for the robot's path error enhancement to some extent. The results show that the path error compensated with the proposed method and method 3 almost falls within the interval of 0.1–0.2 mm, demonstrating the effectiveness of temporal features between path points. Due to the consideration of spatial information of joint configurations, the proposed method exhibits a smaller average error than Method 3. It is observed that, with the combined effect of the proposed HTN and PIC method, the robot's average path error can be reduced to less than 0.2 mm, demonstrating the effectiveness of the proposed compensation method. Furthermore, the path error based on the proposed compensation method is 0.1997 mm and 0.2025 mm, respectively, smaller than other methods after compensation. In summary, the comparative experimental results demonstrate that the proposed compensation method can effectively improve the path accuracy of the industrial robot. Compared with other compensation methods, the proposed compensation method has the lowest path error for the industrial robot, which proves the superiority of this method.

TABLE VII
MILLING PARAMETERS OF THE EXPERIMENT

| Parameter | Value |
|---|---|
| Spindle speed | 7000rpm |
| Thickness of plate | 2mm |
| Material of plate | Aviation aluminum alloy (6061) |
| Feed rate | 100mm/min |
| Diameter of cutter | 10mm |
| Cutting allowance | 2mm |


(a)

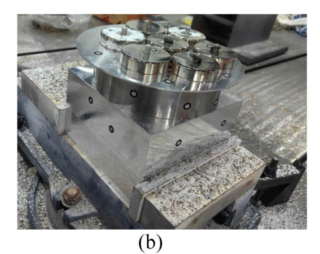

(b)

Fig. 12. Two typical skin parts to be machined. (a) Square skin part. (b) Circular skin part.

### C. Milling Experiment

The proposed error compensation method can be used to operate robotic milling in aircraft skin. Therefore, the milling experiments are designed to verify the milling accuracy of the aircraft skin part. As shown in Fig. 12, two skin parts, including a square skin part sample (Part A) and a circular skin part sample (Part B), are mounted through a fixture and powerful magnet.

Each skin part is machined five times: four times using different path compensation methods, which have been mentioned earlier in path error compensation experiment. Some key milling parameters are tabulated in Table VII. Moreover, all skin parts are measured using a 3-D scanner to obtain the boundary point cloud. Fig. 13 illustrates the robotic milling results part and the skin parts measured boundary point cloud.

In addition, the average contour error is adopted as an index to reflect the overall milling accuracy of each skin part. As shown in Fig. 14, the contour error is defined as the deviation between the measured point cloud of the milling contour and the nearest reference point of reference contour, which can directly affect the form and size accuracy of robot milling, and it can be

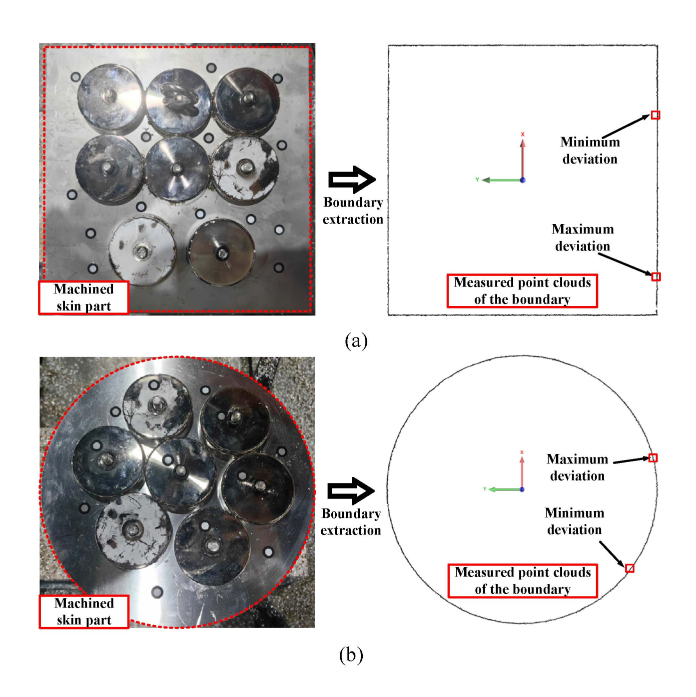



Fig. 13. Machined skin part and the measured boundary point cloud. (a) Square skin part. (b) Circular skin part.

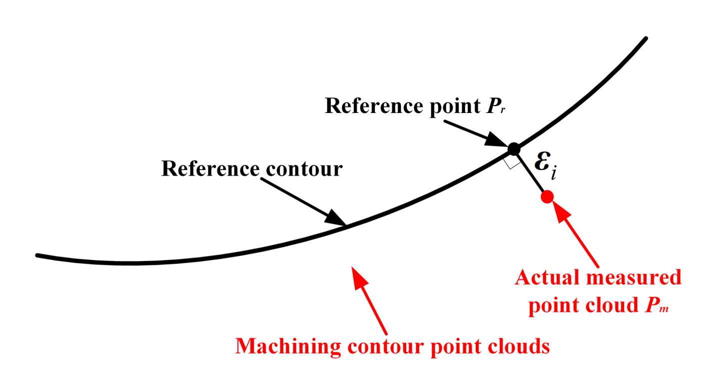



Fig. 14. Definition of contour error.

expressed as follows [6]:

$$\varepsilon_i = \min(P_r - P_m)$$
$$H = \frac{\sum_{i=1}^{N} \|\varepsilon_i\|}{N} \tag{21}$$

where $P_r$ is the reference point of the theoretical skin part model, $P_m$ is the actual milling point cloud and is computed by evaluating the shortest distance between $P_r$ and $P_m$, and $H$ is the average contour error of the skin part. The absolute value deviation between the expected edge contour and the measured actual contour is shown in Fig. 15. The maximum deviation and minimum deviation of the skin part are shown in Table VIII. The contour error of the two skin parts is shown in Table IX.

Fig. 15 shows that the overall error of the compensated group is smaller than that of the uncompensated group. Table VIII indicates that the maximum deviation and minimum deviation of the square skin part are reduced from 0.91 mm to 0.49 mm and 0.35 mm to 0.16 mm, respectively. Furthermore, the maximum

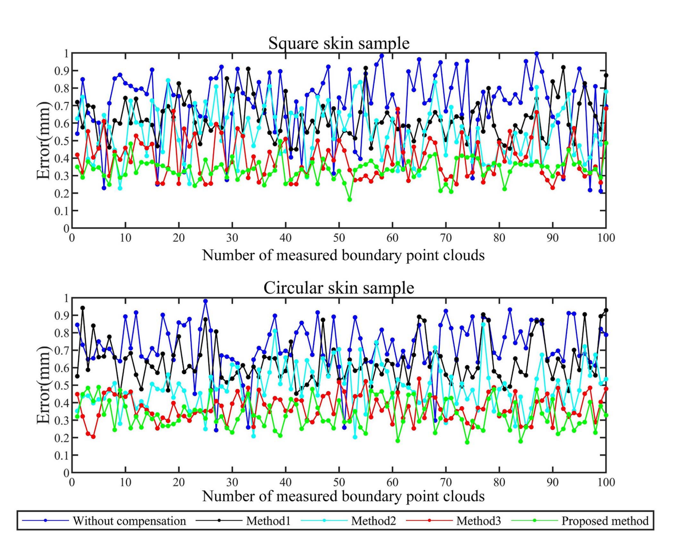



Fig. 15. Results of absolute average error of two skin parts.

TABLE VIII
MAXIMUM AND MINIMUM DEVIATION ERROR OF SKIN PARTS (UNIT: MM)

| Comparison Method | Maximum deviation (Part A) | Minimum deviation (Part A) | Maximum deviation (Part B) | Minimum deviation (Part B) |
|---|---|---|---|---|
| Proposed method | **0.49** | **0.16** | **0.48** | **0.17** |
| Method 3 | 0.50 | 0.23 | 0.54 | 0.20 |
| Method 2 | 0.74 | 0.23 | 0.74 | 0.21 |
| Method 1 | 0.81 | 0.25 | 0.80 | 0.25 |
| Without compensation | 0.91 | 0.35 | 0.92 | 0.45 |

TABLE IX
CONTOUR ERROR OF MACHINED SKIN PART (UNIT: MM)

| | Without compensation | Method1 | Method2 | Method3 | Proposed Method |
|---|---|---|---|---|---|
| Part A | 0.70 | 0.62 | 0.55 | 0.45 | **0.34** |
| Part B | 0.72 | 0.63 | 0.48 | 0.44 | **0.33** |

deviation and minimum deviation of the circular skin part are reduced from 0.92 mm to 0.48 mm and 0.45 mm to 0.17 mm, respectively. Table IX indicates that the contour error of the proposed compensation method is smaller than those of other methods. Specifically, due to the enhanced path accuracy of the robot (resulting in improved end-effector tool coordinate system (TCP) point accuracy), the contour error of the square skin part is reduced from 0.70 to 0.34 mm, and the contour error of the circular skin part is reduced from 0.72 to 0.33 mm. The milling accuracy of the two skin parts is increased by 51.42% and 54.17%, respectively, meeting the milling accuracy requirement of ±0.35 mm average contour error. Therefore, the experimental results show that utilizing the proposed compensation method can reduce the contour error of skin parts in robotic milling.

## V. CONCLUSION

This article proposes a path error compensation method to improve the milling accuracy of aircraft skin parts by using a two-stage compensation method. First, the HTN considers the spatial information of joint configuration and the dependencies of milling path points simultaneously. Therefore, this

proposed HTN is suited for continuous milling operations. Then, to overcome the scarcity of robotic milling datasets, a dataset augmentation method is proposed. This method enriches data diversity by comparing the sequence dispersion of the milling path. In addition, the milling path points are modified to improve the final milling accuracy of the robotic milling system by using the PIC method. A series of experiments are carried out to show the performance of the proposed method. Path error compensation experiment shows that the path error of industrial robots can be effectively reduced with the proposed compensation method. Furthermore, in the robotic milling experiment, the milling results show that the proposed method is of great help in reducing the average contour error of skin parts. In the future, we will continuously carry out further research on robotic layout optimization, online control of robots, and so on.

## References


[1] X. Zhao, B. Tao, S. Han, and H. Ding, “Accuracy analysis in mobile robot machining of large-scale workpiece,” *Robot. Comput.-Integr. Manuf.*, vol. 71, 2021, Art. no. 102153.

[2] G. Wang et al., “Simultaneous calibration of multicoordinates for a dual-robot system by solving the AXB = YCZ problem,” *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1172–1185, Aug. 2021.

[3] D. Gates, “Boeing’s struggle with 777 assembly robots adds to Everett production snarl the Seattle times,” 2016. [Online]. Available: https://www.seattletimes.com/business/boeingaerospace/boeingsstruggle-with-777-assembly-robots-adds-to-everett-production-snarl

[4] R. Müller, M. Esser, and M. Vette, “Reconfigurable handling systems as an enabler for large components in mass customized production,” *J. Intell. Manuf.*, vol. 24, pp. 977–990, 2013.

[5] G. Wang, W.-L. Li, C. Jiang, and H. Ding, “Machining allowance calculation for robotic edge milling an aircraft skin considering the deformation of assembly process,” *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 5, pp. 3350–3361, Oct. 2022.

[6] G. Wang et al., “Trajectory planning and optimization for robotic machining based on measured point cloud,” *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1621–1637, Jun. 2022.

[7] C. Jiang et al., “A novel dual-robot accurate calibration method using convex optimization and lie derivative,” *IEEE Trans. Robot.*, vol. 40, pp. 960–977, 2024, doi: 10.1109/TRO.2023.3344025.

[8] H. Xie, W.-L. Li, D.-H. Zhu, Z.-P. Yin, and H. Ding, “A systematic model of machining error reduction in robotic grinding,” *IEEE/ASME Trans. Mechatron.*, vol. 25, no. 6, pp. 2961–2972, Dec. 2020.

[9] A. Y. Elatta, L. P. Gen, F. L. Zhi, Y. Daoyuan, and L. Fei, “An overview of robot calibration,” *Inf. Technol. J.*, vol. 3, no. 1, pp. 74–78, 2004.

[10] S. Hayati and M. Mirmirani, “Improving the absolute positioning accuracy of robot manipulators,” *J. Robot. Syst.*, vol. 2, no. 4, pp. 397–413, 2004.

[11] J. He, L. Gu, G. Yang, Y. Feng, S. Chen, and Z. Fang, “A local POE-based self-calibration method using position and distance constraints for collaborative robots,” *Robot. Comput.-Integr. Manuf.*, vol. 86, 2024, Art. no. 102685.

[12] G. Alici and B. Shirinzadeh, “Enhanced stiffness modeling, identification and characterization for robot manipulators,” *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 554–564, Aug. 2005.

[13] K. Yang, W. Yang, G. Cheng, and B. Lu, “A new methodology for joint stiffness identification of heavy duty industrial robots with the counterbalancing system,” *Robot. Comput.-Integr. Manuf.*, vol. 53, pp. 58–71, 2018.

[14] M. Cordes and W. Hintze, “Offline simulation of path deviation due to joint compliance and hysteresis for robot machining,” *Int. J. Adv. Manuf. Technol.*, vol. 90, pp. 1075–1083, 2017.

[15] K. Deng, D. Gao, S. Ma, C. Zhao, and Y. Lu, “Elasto-geometrical error and gravity model calibration of an industrial robot using the same optimized configuration set,” *Robot. Comput.-Integr. Manuf.*, vol. 83, 2023, Art. no. 102558.

[16] Y. Song et al., “Industrial serial robot calibration considering geometric and deformation errors,” *Robot. Comput.-Integr. Manuf.*, vol. 76, 2022, Art. no. 102328.

[17] A. Joubair, L. F. Zhao, P. Bigras, and I. Bonev, “Absolute accuracy analysis and improvement of a hybrid 6-DOF medical robot,” *Ind. Robot*, vol. 42, pp. 44–53, 2015.

[18] K. Kamali and I. A. Bonev, “Optimal experiment design for elasto-geometrical calibration of industrial robots,” *IEEE/ASME Trans. Mechatron..*, vol. 24, no. 6, pp. 2733–2744, Dec. 2019.

[19] J. Santolaria, J. Conte, and M. Ginés, “Laser tracker-based kinematic parameter calibration of industrial robots by improved CPA method and active retroreflector,” *Int. J. Adv. Manuf. Technol.*, vol. 66, pp. 2087–2106, 2015.

[20] X. Wei et al., “A fast dynamic pose estimation method for vision-based trajectory tracking control of industrial robots,” *Measurement*, vol. 231, 2024, Art. no. 114506.

[21] Y. Bai, “On the comparison of model-based and modeless robotic calibration based on a fuzzy interpolation method,” *Int. J. Adv. Manuf. Technol.*, vol. 31, no. 11, pp. 1243–1250, 2007.

[22] W. Zhu, W. Qu, L. Cao, D. Yang, and Y. Ke, “An off-line programming system for robotic drilling in aerospace manufacturing,” *Int. J. Adv. Manuf. Technol.*, vol. 68, pp. 2535–2545, 2013.

[23] Y. Zeng, W. Tian, D. Li, X. He, and W. Liao, “An error-similarity-based robot positional accuracy improvement method for a robotic drilling and riveting system,” *Int. J. Adv.Manuf. Technol.*, vol. 88, pp. 2745–2755, 2017.

[24] D. Chen, P. Yuan, T. Wang, C. Ying, and H. Tang, “A compensation method based on error similarity and error correlation to enhance the position accuracy of an aviation drilling robot,” *Meas. Sci. Technol.*, vol. 29, no. 8, 2018, Art. no. 085011.

[25] K. Min, F. Ni, Z. Chen, H. Liu, and C.-H. Lee, “A robot positional error compensation method based on improved kriging interpolation and Kronecker products,” *IEEE Trans. Ind. Electron.*, vol. 71, no. 4, pp. 3884–3893, Apr. 2024.

[26] H.-N. Nguyen, J. Zhou, and H.-J. Kang, “A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network,” *Neurocomputing*, vol. 151, pp. 996–1005, 2015.

[27] P. Yuan, D. Chen, T. Wang, S. Cao, Y. Cai, and L. Xue, “A compensation method based on extreme learning machine to enhance absolute position accuracy for aviation drilling robot,” *Adv. Mech. Eng.*, vol. 10, no. 3, 2018, doi: 10.1177/1687814018763411.

[28] B. Li, W. Tian, C. Zhang, F. Hua, G. Cui, and Y. Li, “Positioning error compensation of an industrial robot using neural networks and experimental study,” *Chin. J. Aeronaut.*, vol. 35, no. 2, pp. 346–360, Feb. 2022.

[29] H. Q. Cao, H. X. Nguyen, T. N.-C. Tran, H. N. Tran, and J. W. Jeon, “A robot calibration method using a neural network based on a butterfly and flower pollination algorithm,” *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 3865–3875, Apr. 2022.

[30] W. Yang, S. Li, Z. Li, and X. Luo, “Highly accurate manipulator calibration via extended Kalman filter-incorporated residual neural network,” *IEEE Trans. Ind. Inform.*, vol. 19, no. 11, pp. 10831–10841, Nov. 2023.

[31] G. Zhao, P. Zhang, G. Ma, and W. Xiao, “System identification of the nonlinear residual errors of an industrial robot using massive measurements,” *Robot. Comput.-Integr. Manuf.*, vol. 59, pp. 104–114, 2019.

[32] W. Wang, W. Tian, W. Liao, B. Li, and J. Hu, “Error compensation of industrial robot based on deep belief network and error similarity,” *Robot. Comput.-Integr. Manuf.*, vol. 73, 2022, Art. no. 102220.

[33] H. Sun, J. Zhang, R. Ma, and X. Zhang, “In-process tool condition forecasting based on a deep learning method,” *Robot. Comput.-Integr. Manuf.*, vol. 64, 2020, Art. no. 101924.

[34] D. K. Bilal, M. Unel, L. T. Tunc, and B. Gonul, “Development of a vision based pose estimation system for robotic machining and improving its accuracy using LSTM neural networks and sparse regression,” *Robot. Comput.-Integr. Manuf.*, vol. 74, 2022, Art. no. 102262.

[35] C. Gong, J. Yuan, and J. Ni, “Nongeometric error identification and compensation for robotic system by inverse calibration,” *Int. J. Mach. Tools Manuf.*, vol. 40, no. 14, pp. 2119–2137, Nov. 2000.

[36] S. Yin, Y. Ren, J. Zhu, S. Yang, and S. Ye, “Vision-based self-calibration method for robotic visual inspection systems,” *Sensors*, vol. 13, no. 12, pp. 16565–16582, 2013.

[37] J. Bi, X. Zhang, H. Yuan, J. Zhang, and M. Zhou, “A hybrid prediction method for realistic network traffic with temporal convolutional network and LSTM,” *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 1869–1879, Jul. 2022.

[38] C. Zhong, G. Li, and Z. Meng, “Beluga whale optimization: A novel nature-inspired metaheuristic algorithm,” *Knowl.-Based Syst.*, vol. 251, 2022, Art. no. 109215.

[39] D. P. Kingma and L. J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representation*, San Diego, CA, USA, 2015, pp. 1–15.

[40] B. Xue, J. Zhu, Z. Zhao, J. Wu, Z. Liu, and Q. Wang, "Validation and mathematical model of workspace measuring and positioning system as an integrated metrology system for improving industrial robot positioning," *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 228, pp. 422–440, 2014.



**Qingyu Peng** received the B.E. degree in marine engineering from the Wuhan University of Technology, Wuhan, China, in 2020. He is currently working toward the Doctoral degree in mechanical engineering with the Huazhong University of Science and Technology, Wuhan.

His research interests include robot machining and robot measurement.



**Wenlong Li** (Member, IEEE) received the B.E. degree in mechanical engineering and automation from Xi'an Jiaotong University, Xi'an, China, in 2004, and the Ph.D. degree in mechatronic engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2010.

He is currently a Professor with HUST, Wuhan. His current research interests include robotic machining and 3-D optical measurement.



**Cheng Jiang** received the B.E. degree in mechanical design, manufacturing and automation in 2016 from the Huazhong University of Science and Technology, Wuhan, China, where he is currently working toward the Doctoral degree in mechanical engineering.

His research interests include point cloud processing and robot machining.



**Yuqi Cheng** received the B.E. degree in mechanical design, manufacturing and automation in 2020 from the Huazhong University of Science and Technology, Wuhan, China. He is currently working toward the Doctoral degree in mechanical engineering with the Huazhong University of Science and Technology, Wuhan.

His research interests include point cloud processing and anomaly detection.



**Dongfang Wang** received the B.E. degree in mechanical design, manufacturing and automation from the Wuhan University of Technology, Wuhan, China, in 2020. He is currently working toward the Doctoral degree in mechanical engineering with the Huazhong University of Science and Technology, Wuhan.

His research interests include robot measurement and machining.



**Wei Xu** (Member, IEEE) received the B.E. and M.Eng. degrees in fluid machinery and engineering from the China University of Mining and Technology, Xuzhou, China, in 2013 and 2016, respectively, and the Ph.D. degree in measurement engineering from Leibniz University, Hannover, Germany, in 2021.

He is a Lecturer with the Huazhong University of Science and Technology, Wuhan, China. His research interests include robotic measurement and machining.